

12/21/99

JC553 U.S. PTO

Frederick P. Fish  
1855-1930

W.K. Richardson  
1859-1951

12-27-99

FISH & RICHARDSON P.C.

4225 Executive Square  
Suite 1400  
La Jolla, California  
92037

December 20, 1999

Telephone  
858 678-5070

Facsimile  
858 678-5099

Attorney Docket No.: 10559/095001/P7611

Web Site  
www.fr.com

Box Patent Application

Assistant Commissioner for Patents

Washington, DC 20231

JC526 U.S. PTO

09/469960

12/21/99

Presented for filing is a new original patent application of:

Applicant: MARK L. SKARPNESS

Title: PRIORITIZING DATA TRANSFERS OVER DATA BUSES

Enclosed are the following papers, including those required to receive a filing date under 37 CFR 1.53(b):

	Pages
Specification	9
Claims	5
Abstract	1
Declaration	[To be Filed at a Later Date]
Drawing(s)	2

Enclosures:

— Postcard.

Basic filing fee	\$0
Total claims in excess of 20 times \$18	\$0
Independent claims in excess of 3 times \$78	\$0
Fee for multiple dependent claims	\$0
Total filing fee:	\$0

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL340524715US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

December 20, 1999

Date of Deposit

Signature

Typed or Printed Name of Person Signing Certificate

BOSTON

DELAWARE

NEW YORK

SILICON VALLEY

SOUTHERN CALIFORNIA

TWIN CITIES

WASHINGTON, DC

FISH & RICHARDSON P.C.

Assistant Commissioner for Patents

December 20, 1999

Page 2

No filing fee is being paid at this time. Please apply any other required fees, **EXCEPT FOR THE FILING FEE**, to deposit account 06-1050, referencing the attorney document number shown above. A duplicate copy of this transmittal letter is attached.

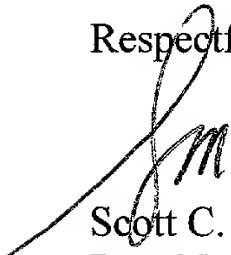
If this application is found to be incomplete, or if a telephone conference would otherwise be helpful, please call the undersigned at (858) 678-5070.

Kindly acknowledge receipt of this application by returning the enclosed postcard.

Please send all correspondence to:

SCOTT C. HARRIS  
Fish & Richardson P.C.  
4225 Executive Square, Suite 1400  
La Jolla, CA 92037

Respectfully submitted,



Scott C. Harris  
Reg. No. 32,030  
Enclosures  
SCH/rpi  
10011781.doc

0443990-1219  
666227-09662450

APPLICATION  
FOR  
UNITED STATES LETTERS PATENT

TITLE: PRIORITIZING DATA TRANSFERS OVER DATA BUSES  
APPLICANT: MARK L. SKARPNESS

0946390-122199

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL340524724US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Date of Deposit December 21, 1999

Signature 

Typed or Printed Name of Person Signing Certificate Deborah Dean

**PRIORITIZING DATA TRANSFERS OVER DATA BUSES**

Technological Field

5           This application relates to data transfers on a data bus, such as the "Universal Serial Bus" (USB) found in many personal computer systems.

Background

10           Data buses are used to transfer data between devices in computer systems. In general, a data bus can transfer only a certain amount of data in a given amount of time and thus has limited "bandwidth." Some data buses, such as a "Universal Serial Bus" (USB), support more than one type of data transfer, each with its own bandwidth and priority requirements. USB, for example, supports two types of streaming transfers for bi-directional data, known as "isochronous" transfers and "bulk" transfers.

15           In the USB environment, isochronous transfers are used for higher priority data, such as packetized voice data, that must reach the intended target within a bounded period of time. Isochronous transfers therefore attempt to guarantee at least some minimum bandwidth on the bus. Bulk transfers, on the other hand, are used with lower priority data, such as normal Internet

20

traffic, for which only a guarantee of delivery is required.  
Bulk transfers do not guarantee bandwidth or priority on the  
bus.

In a conventional computer system, each network-interfacing  
device places all traffic that arrives over the network into a  
single queue. For a USB device, such as a USB broadband modem,  
this means that all incoming data is placed in a single queue,  
regardless of the data's priority. Therefore, the delivery of  
higher priority data is often delayed by the presence of lower  
priority data in the queue. This makes it difficult to provide  
guaranteed bus bandwidth to the higher priority data.

#### Summary

A computer system includes a host processor and a  
peripheral device coupled by an attachment bus. The peripheral  
device transfers data to the host processor over the attachment  
bus using at least two types of data transfers. The peripheral  
device includes a classifying circuit that separates the data  
into a first class associated with a first type of transfer and  
a second class associated with a second type of transfer. The  
peripheral circuit also includes two queues, including a first  
queue that receives the first class of data from the classifying  
circuit and a second queue that receives the second class of  
data from the classifying circuit. A control circuit places

data from the first queue onto the bus with higher priority than data from the second queue is placed onto the bus.

Other embodiments and advantages will become apparent from the following description and from the claims.

5

### Description of the Drawings

FIGS. 1 and 2 are partial block diagrams of a computer system that places incoming and outgoing network traffic into queues that indicate a priority level for each block of data.

### Detailed Description

FIG. 1 shows a computer system 100 in which a host device 105 communicates with a peripheral device 110, such as a broadband modem, over a bi-directional data bus 115, such as a "Universal Serial Bus" (USB) or a "Peripheral Component Interface" (PCI) bus. The peripheral device 110 also connects to a computer network 120, such as the Internet or a private intranet. The peripheral device includes a device-bus interface component 125 that allows the device to send and receive data over the data bus 115. Likewise, the peripheral device 110 includes a network interface component 130 that allows the device to send and receive data over the network 120. The host device 105 includes a host-bus interface 135 that allows the host device 105 to send and receive data over the data bus 115.

Some of the data flowing through the peripheral device 110 has higher priority than other data flowing through the device and therefore must be handled more reliably. For example, packetized voice data must reach its intended target in a manner that has very little, if any, effect on the reproduced sound quality. Normal Internet traffic, on the other hand, generally does not need to reach its intended target with any particular urgency.

When the data bus 115 is a USB bus, isochronous transfers, or "guaranteed service" transfers, are used to deliver higher priority data, such as packetized voice data, between the peripheral device 110 and the host processor. USB uses bulk transfers, or "best effort service" transfers, to deliver lower priority data, such as normal Internet traffic.

The host device 105 and the peripheral device 110 both include components that distinguish between higher priority data and lower priority data. These components ensure that the higher priority data is delivered with greater reliability.

The host device 105 typically includes the computer's base board, on which a host processor 140 and host memory device 145 reside. The host processor 140 executes a variety of application programs, including applications that require guaranteed service, such as packetized voice applications 150, and programs that require only best-effort service, such as

Internet browsing applications 155. The host processor 140 also implements a traffic shaping component 160, which receives data addressed to the network and separates the data into higher priority and lower priority blocks. The host processor 140  
5 implements the application programs 150, 155 and the traffic shaping component 160 by retrieving executable instructions from a storage device, such as the host memory device 145, and executing these instructions.

10 The host memory device 145 includes storage elements that are partitioned to form two or more data queues 165, 170, 175. At least one queue is a "highest priority" queue 165 dedicated to storing data generated by or intended for one of the higher priority applications 150, and at least one queue is a "low priority" queue 175 dedicated to storing data generated by or intended for the lower priority applications 155. In some embodiments, one queue is an "elevated priority" queue 170 dedicated to storing data that is high priority, but that is not as high priority as the data stored in the highest priority queue 165.

20 For example, in some embodiments the highest priority queue 165 stores data that must be delivered at a constant bit rate. This data requires the guarantee of a fixed bus bandwidth. The elevated priority queue 170 stores data that can be delivered at a variable, but bounded, bit rate. This data requires the



guarantee of some minimum bus bandwidth. The low priority queue 175 stores data that can be delivered at an unspecified bit rate and thus requires no guarantee of bus bandwidth.

5 The peripheral device 110 also includes at least two queues, including a "higher priority" queue 180 that stores higher priority data and a "lower priority" queue 185 that stores lower priority data. Data flowing into the peripheral device 110 through the network interface component 120 passes through a traffic classifier component 190. This component separates the incoming data blocks into higher priority and lower priority streams and feeds the streams to the appropriate queues. The device-bus interface 125 places data from the queues 180, 185 onto the data bus 115 in a manner that honors the bandwidth and priority requirements of the data. In general, a minimum amount of data is pulled from the higher priority queue 180 during each time cycle of some predefined length.

10  
15  
20 Data flowing into the peripheral device through the device-bus interface 125 arrives from the host device 105. In some embodiments, the peripheral device includes another traffic classifier between the device-bus interface 125 and the queues 180, 185. In most embodiments, however, the host-bus interface 135 in the host device 105 produces a signal indicating the priority of each block of data, so a traffic classifier is not

needed between the device-bus interface 125 and the queues 180, 185. The device-bus interface 125 simply directs the incoming data blocks to the appropriate queues based upon the content of this signal.

5        FIG. 2 shows one implementation of the traffic classifier component 190 in the peripheral device. This implementation is designed for use with an "Asynchronous Transfer Mode" (ATM) network and a "Universal Serial Bus" (USB).

10        Data arriving through the network interface component 130 (FIG. 1) enters a first-in-first-out (FIFO) buffer 200 that holds one data packet, or "cell." In an ATM environment, each cell includes 53 bytes of data arranged in a commonly-defined configuration. Of the 53 bytes in each cell, four bytes carry a header that, among other things, indicates the packet's virtual channel identifier and Virtual Path Identifier (VCI:VPI). The VCI:VPI typically occupies bits 4-27 in an ATM cell.

15        In general, all ATM data cells at a particular priority level are transferred over a known set of virtual channels. Therefore, the peripheral device can derive each cell's priority  
20        level based upon the VCI:VPI encoded in the cell's header. The priority level associated with each virtual channel is stored in a content addressable memory (CAM) device 205 in the peripheral device.

The FIFO buffer 200 transfers the incoming data, one bit at a time, into a four-byte shift register 210. Therefore, in an ATM environment, bits 4-27 in the shift register contain a VCI:VPI once every 53 bytes. The CAM device 205 is programmed to retrieve the VCI:VPI from each data packet by loading these 24 bits once every 53 bytes. Upon loading the VCI:VPI from a data packet, the CAM device 205 compares the packet's VCI:VPI to each of the stored VCI:VPI's that are associated with higher priority cells. If the packet's VCI:VPI matches one of the stored VCI:VPI's, the CAM device 205 enables the higher priority queue 180 and disables the lower priority queue 185, so the incoming packet is stored in the higher priority queue 180. If the packet's VCI:VPI does not match any of the stored packets, the CAM device enables the lower priority queue 185 and disables the higher priority queue 180, and the packet is stored in the lower priority queue 185.

A device controller 215 in the device-bus interface component 125 (FIG. 1) controls the flow of data from the queues 180, 185 to the data bus. The device controller 215 awards control of the USB to the higher priority queue 180 once each time cycle, e.g., once every millisecond. The device controller 215 awards control of the data bus to the lower priority queue 185 only when the bus is otherwise unoccupied. In this manner,

the peripheral device guarantees that higher priority data is delivered at some minimum rate.

A number of embodiments of the invention are described above. A person of ordinary skill will recognize that various  
5 modifications are possible without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. A computer system comprising:

a host processor;

5 a peripheral device configured to transfer data to the host processor over an attachment bus using at least first and second types of data transfers, comprising:

10 a classifying circuit configured to separate the data into a first class associated with the first type of transfer and a second class associated with the second type of transfer;

15 a first queue connected to receive the first class of data from the classifying circuit;

20 a second queue connected to receive the second class of data from the classifying circuit; and

25 a control circuit configured to place data from the first queue onto the bus at a higher priority than data from the second queue is placed onto the bus.

30 2. The system of claim 1, where the bus is configured to receive data during time periods of predetermined length, and where the controller is configured to place at least a minimum amount of data from the first queue onto the bus during each time period.

3. The system of claim 1, where the controller is configured to place data from the second queue onto the bus only when the bus is otherwise unoccupied.

5

4. The system of claim 1, where the peripheral device includes a network interface component connected to receive the data from a computer network.

5. The system of claim 4, wherein the data includes packetized voice data.

6. The system of claim 1, where the bus is a Universal Serial Bus and the peripheral device is configured to transfer data over the bus using isochronous and bulk transfers.

7. The system of claim 1, where the peripheral device is configured to deliver the data in packets of predetermined length.

20

8. The system of claim 7, where the classifying circuit is configured to place each of the packets into one of the classes.

9. The system of claim 7, where a portion of each packet indicates the type of transfer associated with the packet, and where the classifying circuit includes a storage device that stores information indicating each of the types of transfer that is associated with at least one of the classes.

10. The system of claim 9, where the classifying circuit includes a selection element configured to compare, for each packet, the information in the storage device to the data in the portion of the packet that indicates the type of transfer and configured to select a corresponding one of the queues to receive the packet.

11. A method comprising:

transferring data to a host processor over an attachment bus using at least first and second types of data transfers and, in transferring the data:

separating the data into a first class associated with the first type of transfer and a second class associated with the second type of transfer; and

placing data of the first class onto the bus at a higher priority than data of the second class is placed onto the bus.

12. The method of claim 11, further comprising placing  
data on the bus during time periods of predetermined length,  
where placing data of the first class on the bus includes  
placing at least a minimum amount of data of the first class  
5 onto the bus during each time period.

13. The method of claim 11, further comprising placing  
data of the second class onto the bus only when the bus is  
otherwise unoccupied.

14. The method of claim 11, further comprising receiving  
the data from a computer network.

15. The method of claim 14, where receiving the data  
includes receiving packetized voice data.

16. The method of claim 11, where transferring data  
includes delivering the data in packets of predetermined length.

17. The method of claim 16, where separating the data  
includes placing each of the packets into one of the classes.



18. The method of claim 16, further comprising storing information indicating each of the types of transfer that is associated with at least one of the classes.

5 19. The method of claim 18, further comprising:  
comparing, for each packet, the stored information to a portion of the data in the packet that indicates the type of transfer associated with the packet; and  
placing the packet into a corresponding one of the classes.

Abstract

A computer system includes a host processor and a peripheral device coupled by an attachment bus. The peripheral device transfers data to the host processor over the attachment bus using at least two types of data transfers. The peripheral device includes a classifying circuit that separates the data into a first class associated with a first type of transfer and a second class associated with a second type of transfer. The peripheral circuit also includes two queues, including a first queue that receives the first class of data from the classifying circuit and a second queue that receives the second class of data from the classifying circuit. A control circuit places data from the first queue onto the bus at a higher priority than data from the second queue is placed onto the bus.

03469960-102499

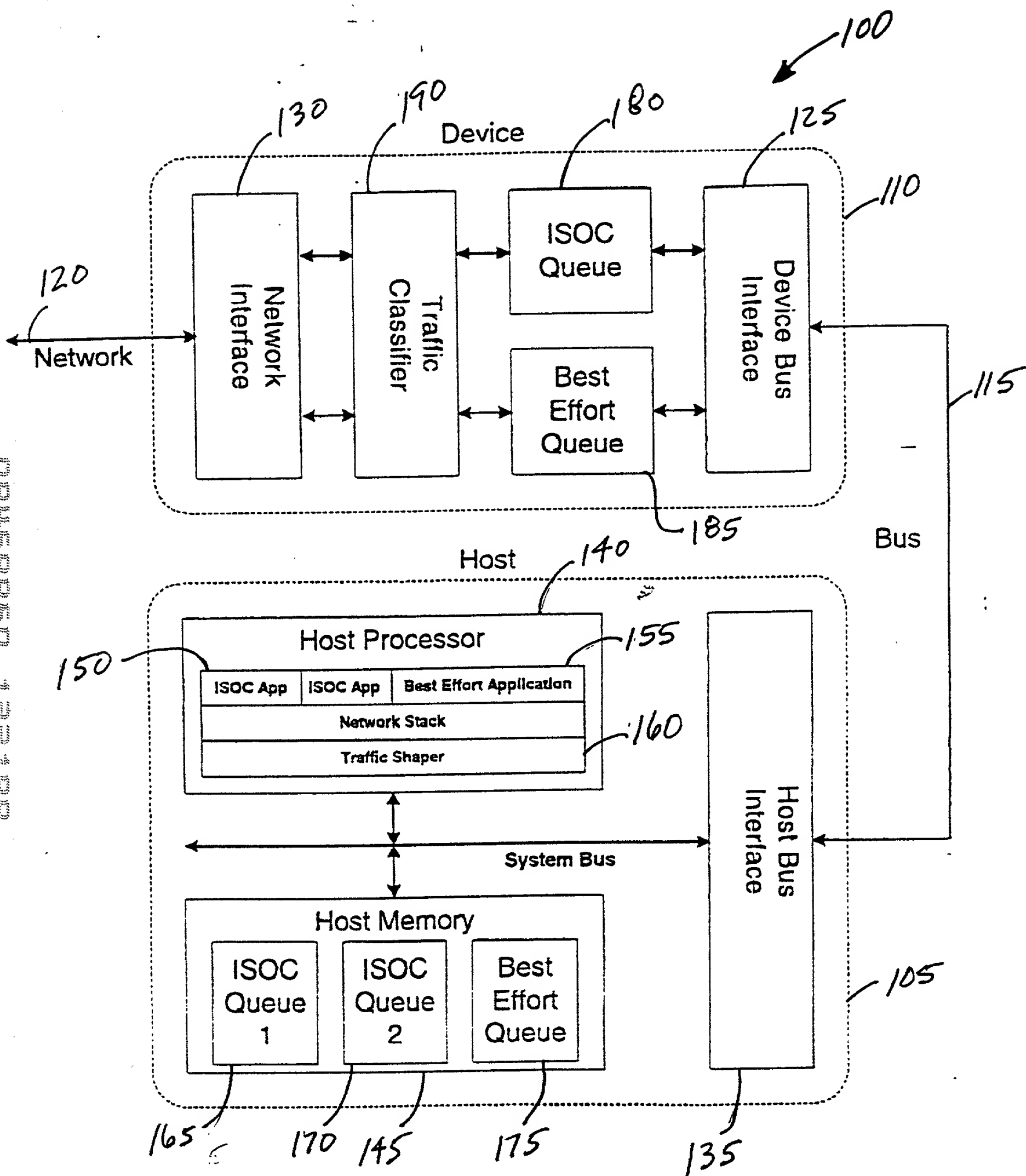


FIGURE 1

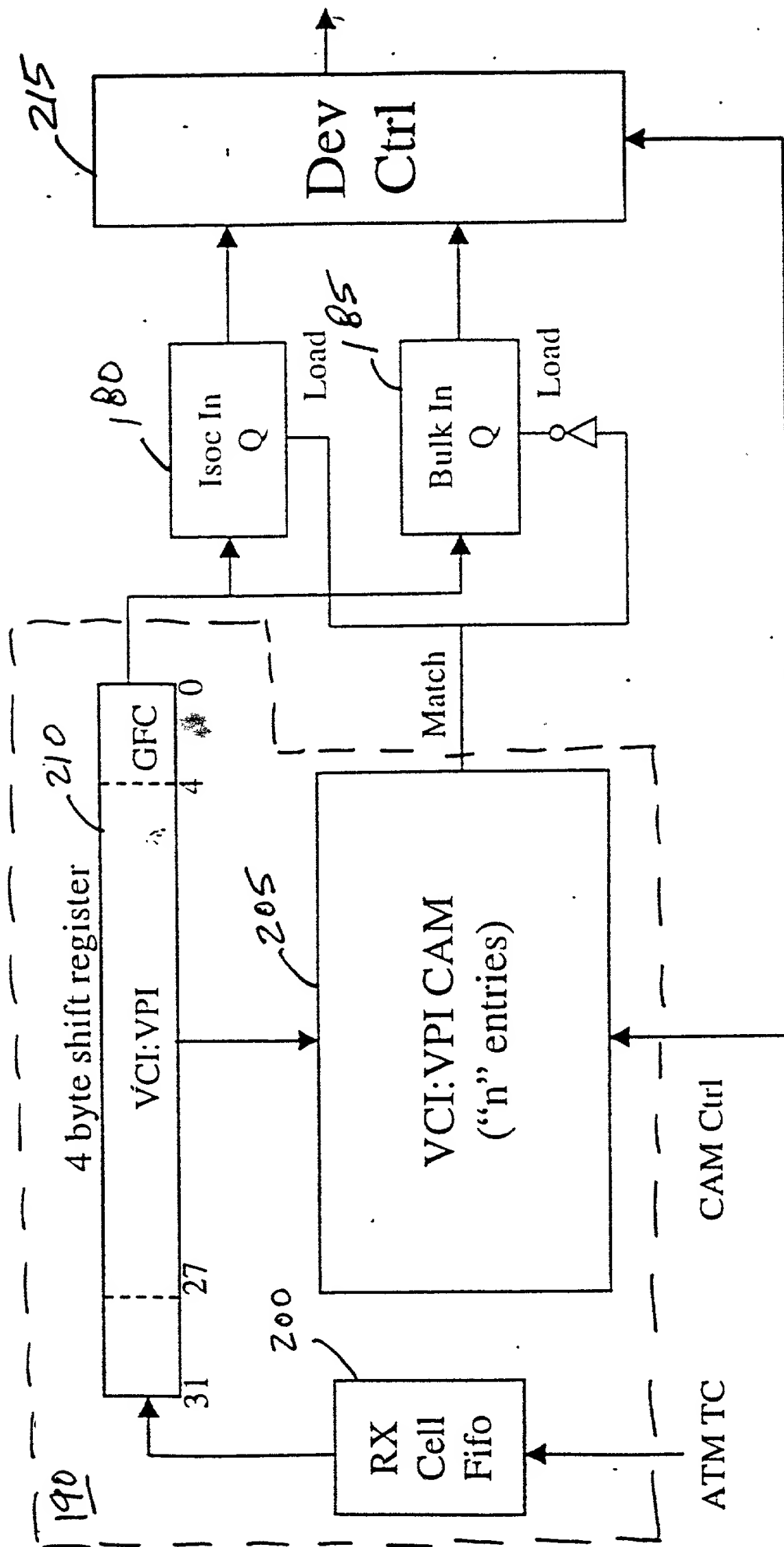


FIGURE 2